Adaptation in P300 braincomputer interfaces: A two-classifier cotraining approach

Panicker, Rajesh C.; Sun, Ying; Puthusserypady, Sadasivan

Published in:
I E E E Transactions on Biomedical Engineering

Link to article, DOI:
10.1109/TBME.2010.2058804

Publication date:
2010

Document Version
Publisher's PDF, also known as Version of record

Adaptation in P300 Brain–Computer Interfaces: A Two-Classifier Cotraining Approach

Rajesh C. Panicker, *Student Member, IEEE*, Sadasivan Puthusserypady*, *Senior Member, IEEE*, and Ying Sun, *Member, IEEE*

***Abstract*—A cotraining-based approach is introduced for constructing high-performance classifiers for P300-based brain–computer interfaces (BCIs), which were trained from very little data. It uses two classifiers: Fisher's linear discriminant analysis and Bayesian linear discriminant analysis progressively teaching each other to build a final classifier, which is robust and able to learn effectively from unlabeled data. Detailed analysis of the performance is carried out through extensive cross-validations, and it is shown that the proposed approach is able to build high-performance classifiers from just a few minutes of labeled data and by making efficient use of unlabeled data. An average bit rate of more than 37 bits/min was achieved with just one and a half minutes of training, achieving an increase of about 17 bits/min compared to the fully supervised classification in one of the configurations. This performance improvement is shown to be even more significant in cases where the training data as well as the number of trials that are averaged for detection of a character is low, both of which are desired operational characteristics of a practical BCI system. Moreover, the proposed method outperforms the self-training-based approaches where the confident predictions of a classifier is used to retrain itself.**

***Index Terms*—Brain-computer interface (BCI), cotraining, EEG, P300, semisupervised learning.**

I. Introduction

A BRAIN–COMPUTER interface (BCI) is a system capable of utilizing the brain's electrical signals for direct communication with a computer system, without reliance on the usual neuromuscular pathways. Patterns in the electrical activity of the brain are extracted and used as control signals to the computer or a prosthetic device. A primary target beneficiary group for BCIs are people with severely impaired motor systems, which have few options to communicate with the outside world. There are various BCI systems based on evoked potentials (EPs), event related potentials (ERP), motor imagery, various band rhythms, etc., [1]. The most practical modality for a BCI is the electrical activity measured at the scalp, the EEG. A widely used ERP in practical BCI systems is the P300 [1]. It is produced in response to rare, random, and task-relevant stimulus, commonly known as the oddball paradigm [2]. The P300 usually manifests as a positive peak about 300 ms after the presentation of the stimulus, predominantly at the centro–parietal region [3]. Its amplitude and latency varies from person to person and also depends on the "surprise" in the stimulus [4]. The advantages of P300 include its suitability for a wide spectrum of users including the locked-in patients [5], relaxed requirement of visual attention and relative ease of detection, and reasonably good information transfer rates (ITRs) [6].

A widely used P300 BCI paradigm is the P300 speller [7]. It has characters displayed as a matrix, with rows and columns being highlighted at a pseudorandom sequence. P300 responses are elicited during the highlighting of both the row and the column corresponding to the character of interest. The fact that this elicitation is time locked to the stimulus can be utilized to estimate the row and column containing the character. The presence or absence of P300 is typically detected with the help of a classifier, which is trained using some data for which the labels are known [8], [9]. Given the inter- and intrapersonal variations in EEG, the training time is several tens of minutes to obtain satisfactory performance [1]. The requirement of longer training data is tiring on the part of the user and reduces the attractiveness of BCI as an alternate channel of communication. Hence, building good classifiers from shorter training sessions have become a topic of great interest to the BCI research community. One way to improve the performance is to have improved feature extraction and classification techniques, and the other is to render the classifier to be able to adjust itself progressively as more and more (new) unlabeled data arrive. Such an adaptive BCI has become an active field of research and encouraging results have been reported by various groups [10]–[13].

The labels of the new incoming data are not available for adaptation of the classifiers in BCIs, unlike in active learning scenarios [14]. This necessitates the classifier being able to adapt the classification boundary blindly from the incoming data. Transductive and semisupervised algorithms have been recently used as alternatives to the strenuous training effort required on the part of the user. Transductive algorithms classify the unlabeled data by optimizing a joint function of labeled and unlabeled data. A transductive version of support vector machines (SVMs), which aligns the classification boundary maximally away from the unlabeled data has been proposed for use in BCI systems [15]. Unlike a standard SVM, the optimization problem for transductive SVM is nonconvex. This requires complex numerical routines, and there is no guarantee of the solution being a global

Manuscript received February 10, 2010; revised May 4, 2010; accepted June 16, 2010. Date of publication July 15, 2010; date of current version November 17, 2010. *Asterisk indicates corresponding author*.

R. C. Panicker and Y. Sun are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576, Singapore (e-mail: rajesh.c@nus.edu.sg; elesuny@nus.edu.sg).

*S. Puthusserypady is with the Department of Electrical Engineering, Technical University of Denmark, Lyngby 2800, Denmark (e-mail: spu@elektro.dtu.dk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBME.2010.2058804

optimum. On the other hand, usual semisupervised algorithms define a classifier as a function for which an unlabeled datum is essentially test data—the posterior probabilities of data being labeled are independent. A widely used semisupervised technique is self-training, which uses the most confident predictions from the classifier for additional labeled data [16]. Another popular method is the classical cotraining introduced by Blum and Mitchell [17], which requires two redundant and sufficient views of the data, i.e., two sets of independent features both of which have the classification information. Such a requirement cannot be met in most practical scenarios, including BCI. Goldman and Zhou [18] later showed that labels generated from two different classifiers can also be used to generate additional data, thus doing away with the multiview requirement. Such a two-classifier cotraining-based approach is introduced in this paper to reduce the training effort. The two classifiers used are the Fisher's linear discriminant analysis (FLDA) and the Bayesian linear discriminant analysis (BLDA). The algorithm exploits the difference between the classifiers to generate different labels for the data. Recently, a mathematical reasoning for the success of cotraining style algorithms was given by Wang and Zhou [19]. They proved that the success of cotraining-based algorithms is higher when the difference between the classifiers is maximized [20].

The proposed method is described in Section II. Section III details the experiments and data analysis, followed by discussions in Section IV. The paper is concluded with some remarks in Section V.

II. Cotraining Method

Let the data from the jth trial, $\mathbf{x}_j = [x_{1j}, x_{2j}, \ldots, x_{dj}]^T$ be the feature vector of length d for the classification problem, with x_{ij}s, $i = 1, 2, \ldots, d$, denoting the individual features and let $y_j \in \{-1, 1\}$ be the corresponding labels. Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l, \mathbf{x}_{l+1}, \ldots, \mathbf{x}_n\}$ be the set of all n data points in feature space, of which l points have known labels given by, $Y = \{y_1, y_2, \ldots, y_l\}$. The semisupervised classification problem can be defined as follows: Given the dataset $S = L \cup U$, where $L = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\} \subset X \times Y$ is the labeled dataset and $U = \{\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \ldots, \mathbf{x}_n\} \subset X$ is the unlabeled dataset, find a mapping $h^* \in H$, which holds for the entire S and gives a perfect generalization, where H: $X \rightarrow Y$ denotes the set of all classifiers. This will be hard to realize in most practical applications, where the data are generally noisy; also for small l, the mapping will be less accurate. Cotraining method uses two initial classifiers, namely, $h_1^0 \in H$ and $h_2^0 \in H$, trained on L, and iteratively updates them, with h_1^{i+1} and h_2^{i+1} hopefully providing a better mapping than h_1^i and h_2^i, where i is the iteration number.

The algorithm can be summarized as follows. Given the initial training data (L) and the unlabeled data (U):

1) Obtain the initial classifiers h_1^0 and h_2^0, using the training data $L_1^0 = L_2^0 = L$; and set $i = 1$.
2) Take u number of unlabeled instances from U and label them using h_1^{i-1} and h_2^{i-1}.
3) Construct a new labeled set L_1^i by combining L_1^{i-1} and the labeled data given by h_2^{i-1}, and L_2^i by combining L_2^{i-1} and the labeled data given by h_1^{i-1} in the previous step.
4) Obtain the updated classifiers h_1^i and h_2^i using L_1^i and L_2^i. In certain cases, only a fraction of the most confident among the u labels predicted by each classifier is used for updating.
5) Increment i and repeat steps 2 to 5 till stopping criterion is met.
6) Stop the training if all the unlabeled data have been classified or if the confidence improvement due to the addition of unlabeled data are minimal.

Several classifiers for classification of P300 have been reported in the literature, which include FLDA [21], SVM [22], BLDA [23], [24], etc. The classifiers used (for implementing h_1^i's and h_2^i's, respectively) are BLDA and FLDA for the following reasons.

1) In our preliminary experiments, BLDA and FLDA gave very good accuracies. Some studies have reported that the algorithms give accuracies comparable to that of SVMs [25].
2) Both are computationally simple and do not require complex cross-validation procedures for tuning their hyperparameters. Although BLDA uses a data-dependant expectation-maximization type algorithm for hyperparameter optimization, the empirical complexity was found to be very low, especially as compared to competing classifiers like SVMs.
3) The two classifiers gave reasonably different separating planes, which is a crucial factor in cotraining approaches. Since BLDA uses an entirely different optimization method, the biases of the two algorithms are different. Such a diversity is crucial from the point of cotraining. In our experiments, it was observed that reasonable diversity was maintained, though it is inevitable that the classifiers produce closer and closer predictions as the cotraining proceeds with more and more unlabeled samples.

A brief description of the two algorithms is given shortly.

A. Fisher's Linear Discriminant Analysis

In FLDA, the data are projected to a lower dimension such that the projected means of classes are far apart, while the spread of the projected data is small. This can be realized by optimizing a cost function related to the within-class matrix ($\mathbf{S}_w$) and the between-class matrix ($\mathbf{S}_b$), which are defined as

$$\mathbf{S}_w = \sum_{k=1}^{n^c} \sum_{\mathbf{x}_j \in c_k} (\mathbf{x}_j - \mathbf{m}_k^f)(\mathbf{x}_j - \mathbf{m}_k^f)^T \tag{1}$$

$$\mathbf{S}_b = \sum_{k=1}^{n^c} n_k^c (\mathbf{m}_k^f - \mathbf{m}^f)(\mathbf{m}_k^f - \mathbf{m}^f)^T \tag{2}$$

where $\mathbf{x}_j$, $j = 1, 2, \ldots, n^t$ are the training data vectors, c_k denotes the kth class, $\mathbf{m}_k^f$ is the mean of samples belonging to the kth class, $\mathbf{m}^f$ is the global mean, n^c is the number of classes ($n^c = 2$ in our classification, denoting either the presence or the

absence of P300), and n_k^c is the number of samples in the kth class. Given the pattern matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n^t}]$ and the corresponding label vector $\mathbf{y} = [y_1, y_2, \ldots, y_{n^t}]$, the problem is to find a projection vector $\mathbf{w} = [w_1, w_2, \ldots, w_d]^T$ such that the projection

$$\mathbf{y} = \mathbf{w}^T \mathbf{X} \tag{3}$$

maximizes the criterion function $J_p(\mathbf{w})$ defined as

$$J_p(\mathbf{w}) = \frac{\det(\mathbf{w}^T \mathbf{S}_b \mathbf{w})}{\det(\mathbf{w}^T \mathbf{S}_w \mathbf{w})}. \tag{4}$$

The solution [26] is to choose $\mathbf{w}$ satisfying the eigen equation

$$\mathbf{S}_w^{-1} \mathbf{S}_b \mathbf{w} = \lambda \mathbf{w} \tag{5}$$

if $\mathbf{S}_w^{-1}$ exists, λ being the only nonzero eigenvalue [26] of $\mathbf{S}_w^{-1}\mathbf{S}_b$. Once $\mathbf{w}$ is estimated, the classifier design is complete and the output for a single feature vector ($\mathbf{x}_j$) is

$$y_j = \mathbf{w}^T \mathbf{x}_j. \tag{6}$$

Reliable detection of the P300 usually requires several rounds (a *round* is the data associated with one complete cycle of row and column flashings [27]) of stimulus presentations. In each round, the scores for all rows and columns are calculated using (6). As reliable detection of the P300 requires data from several rounds, the scores are averaged over a fixed number of rounds (denoted by n^R). The estimated target is the symbol at the intersection of the row and the column having the maximum of the averaged scores. This scheme performs a multiclass classification, even though the underlying classifications are binary.

B. Bayesian Linear Discriminant Analysis

The BLDA uses an entirely different approach for optimizing the weights. Instead of committing a particular value of the projection vector, it creates the posterior distribution using the Bayesian criterion. The BLDA implemented in this paper is similar to the one described in [23]. More general descriptions of this method can be found in [28] and [29]. The basic assumption in BLDA is that the regression targets

$$\mathbf{y} = \mathbf{w}^T \mathbf{X} + \mathbf{n} \tag{7}$$

where $\mathbf{n}$ is the noise vector. For simplicity and mathematical tractability, the noise is assumed to be Gaussian. Therefore, the likelihood function can be written as

$$p(D|\beta, \mathbf{w}) = \left(\frac{\beta}{2\pi}\right)^{\frac{l}{2}} e^{-\frac{\beta}{2}\|\mathbf{w}^T \mathbf{X} - \mathbf{y}\|^2} \tag{8}$$

where D denotes the pair $(\mathbf{X}, \mathbf{y})$, β denotes the inverse variance of noise, and l is the number of examples in the training set. For Bayesian inference, we specify a prior distribution for the weight vector $\mathbf{w}$. The expression for the prior distribution (which is assumed to be Gaussian) is

$$p(\mathbf{w}/\alpha) = \prod_{i=1}^{d} \left(\frac{\alpha_i}{2\pi}\right)^{\frac{1}{2}} e^{-\frac{1}{2}\left(\mathbf{w}^T \mathbf{I}'(\alpha) \mathbf{w}\right)} \tag{9}$$

where α_is are the hyperparameters (which signifies the inverse relevance of each feature), and $\mathbf{I}'(\alpha)$ is a $d \times d$ dimensional square matrix, with α_is along the diagonal.

Using Bayes theorem, it can be shown [23] that the posterior is also a Gaussian, with mean ($\mathbf{m}$) and the covariance ($\mathbf{C}$) given by

$$\mathbf{C} = \beta \left(\beta \mathbf{X}\mathbf{X}^T + \mathbf{I}'(\alpha)\right)^{-1} \tag{10}$$

$$\mathbf{m} = \beta \mathbf{C} \mathbf{X} \mathbf{y}. \tag{10b}$$

The predictive distribution of the target y' for previously unseen $\mathbf{x}'$ is also Gaussian, the mean (μ) and variance (σ^2) of which is given by

$$\mu = \mathbf{m}^T \mathbf{x}' \tag{11a}$$

$$\sigma^2 = \frac{1}{\beta} + \mathbf{x}'^T \mathbf{C} \mathbf{x}' \tag{11b}$$

of which only the mean is being used for the class predictions. The likelihood $p(D|\beta, \alpha)$ is given by marginalizing (8) as

$$p(D|\beta, \alpha) = \int p(D|\beta, \mathbf{w}) p(\mathbf{w}|\alpha) d\mathbf{w}. \tag{12}$$

The update equations for α and β are obtained by maximizing the log likelihood and setting the partial derivatives with respect to α and β to 0 as

$$\alpha_i = \frac{1}{c_{ii} + m_i^2} \tag{13a}$$

$$\beta = \frac{d}{\mathrm{tr}(\mathbf{X}\mathbf{X}^T)\mathbf{C} + \|\mathbf{m}^T \mathbf{X} - \mathbf{y}\|^2} \tag{13b}$$

where c_{ii}s are the diagonal elements of $\mathbf{C}$, m_is are the elements of $\mathbf{m}$, and $\mathrm{tr}(\cdot)$ denotes the trace of matrix. Equations (10a), (10c) and (13a), (13c) form a set of coupled equations, which can be iterated to optimize the values of α and β. Once the optimization is complete, mean of the posterior ($\mathbf{m}$) given by (10c) is taken as the optimum value of $\mathbf{w}$. The character detection then proceeds in a manner similar to that with FLDA.

Depending on the semisupervised strategy employed and the classifier giving the final output, we have the following four different classifiers:

1) self-training BLDA (SBLDA);
2) self-training FLDA (SLDA);
3) cotraining BLDA (CBLDA)—in which the output is taken from BLDA classifier, which is cotrained with FLDA;
4) cotraining FLDA (CLDA)—in which the output is taken from FLDA classifier which is cotrained with BLDA.

Performance analysis of these four algorithms is given in Section IV.

C. Confidence Criterion

The cotraining process is repeated once 100 rounds of fresh unlabeled data are made available to the classifier, as it was empirically found to give reasonably good performances while avoiding frequent updating of the classifier. The fraction of the classified data that is being added to the training data in each iteration of the algorithm is determined by a confidence

criterion. In our study, the z-score [defined in (14] is the metric used for calculating the confidence of predictions. The z-score corresponding to the ith character is given as

$$z_i = \frac{y_{\max,i} - y_{\text{mean},i}}{\sigma_{y_i}} \tag{14}$$

where $y_{\max,i}$, $y_{\text{mean},i}$, and σ_{y_i} are the maximum, mean, and the standard deviation, respectively, of the averaged scores corresponding to the rows/columns associated with the ith character detection.

D. Evaluation Criterion

The classification accuracy (CA) is the ratio of correct character classifications to the total number of characters being classified. However, since BCI is a communication system, the ITR is also an important figure of merit. Based on the suggestion of Wolpaw *et al.* [30], the formula for information per detected symbol is calculated as

$$\text{B[bits]} = \log_2(n^s) + p_0 \log_2(p_0) + (1-p_0)\log_2\left[\frac{(1-\text{CA})}{(n^s-1)}\right] \tag{15}$$

where n^s is the number of equiprobable symbols (36 in our speller paradigm; as there are six rows and six columns). It is assumed that CA is uniform among classes. Given the interstimulus interval (ISI, the interval between two consecutive stimulus presentations, in seconds) and the intercharacter gap (ICG, the time gap between two consecutive rounds, in seconds), ITR (in bits/min) is calculated as

$$\text{ITR} = \frac{\text{B[bits]}}{n^R \times \text{ISI} \times 12 + \text{ICG}} \times 60 \tag{16}$$

where ISI and ICG are given in seconds. The CA by chance for all the cases is 0.0278 (1/36, or 2.78%), and the corresponding ITR is 0.

For all the comparisons done in this paper, the sign test is used, given by

$$n_{\text{true}} = \frac{1}{2}\left(n_{\text{iter}} - \sum_{i=1}^{n_{\text{iter}}} sgn\left(\text{CA}_1^i - \text{CA}_2^i\right)\right) \tag{17}$$

where CA_1^i and CA_2^i are the classification accuracies for method 1 and method 2 respectively and n_{iter} is the number of cross-validation iterations. The one-tailed p-value [31] (for the null hypothesis that method 1 does not give better CA than method 2) is calculated using the binomial cumulative distribution function, with n_{true} trials turning out to be true, out of the n_{iter} binomial trials.

III. Data Recording and Analysis

A. Experimental Setup

The experimental setup makes use of a 24-channel amplifier from ANT-Neuro [32], with a sampling rate of 256 Hz. EEG from seven channels (Cz, C3, C4, Pz, P3, P4, and Oz) were recorded following the standard 10–20 system [33]. Electrode AFz was used as the ground, and linked-ear is used as the reference. All electrodes used were passive and unshielded, and the impedances were kept below 10 kΩ throughout the experiment. The experiment was conducted in a laboratory environment, with sound absorbent screens to enable the user to concentrate better, and without electromagnetic shielding. The data recording is controlled from a custom multithreaded program implemented in Visual C++, where one thread is used for data collection and is precisely synchronized with a second thread, which controls an OpenGL-based hardware-accelerated speller interface.

B. Offline Experiments

To evaluate the performance of the proposed scheme, offline experiments were conducted on five healthy subjects aged 22–27; four males and one female. Subjects 1 and 4 had some prior experience with P300 BCIs, whereas the other three were BCI-naive. Each subject performed an experiment of 72 characters, each repeated for 20 rounds, with an ISI of 175 ms. An ICG of one second was provided to enable the subject to shift his/her attention to the next character. In our experiments, the target character was highlighted so that the user does not have to memorize any character order. It also helps minimize the possibility of character positional biases in P300 signal by allowing the usage of random characters as targets.

C. Preprocessing

As most of the discriminant information in P300 resides in lower frequencies, the collected data are zero-phase (forward-backward) bandpass filtered between 0.5 and 12 Hz, using a Butterworth filter of order 3. To reduce the feature size, it is downsampled to 32 Hz, and the data for a duration of 0.7 s (23 samples) from the start of the stimulus are considered to belong to that particular epoch. A 161-dimensional feature vector is constructed by the concatenation of the data thus obtained, from all the seven channels. The optimum number of rounds to be chosen is usually a tradeoff between the CA and the ITR and varies from person to person. In our study, we carried out the analysis using different number of rounds ($n^R = 1$ and 2).

D. Cross-Validation

To evaluate the performance of cotraining, an extensive cross-validation analysis is carried out. First, the data are shuffled 100 (n_{iter}) times randomly with the constraint that the data for any one character are kept together, to obtain 100 different data ensembles. For each ensemble, first l rounds of training data are used to train an initial classifier. The rest of the data are treated as unlabeled data and are progressively added in batches of 100, and the self/cotraining algorithms are applied. The means, standard deviations and p-values are calculated as appropriate, using the results from the 100 ensembles. This scheme gives us a realistic measure of the performance of the algorithm and has been used in many previous works involving semisupervised learning [20]. The disadvantage of such a scheme is that the scrambling of data forces the algorithm to ignore any adaptation effects. However, taking it into consideration would make the data requirement for performance analysis impractically huge.

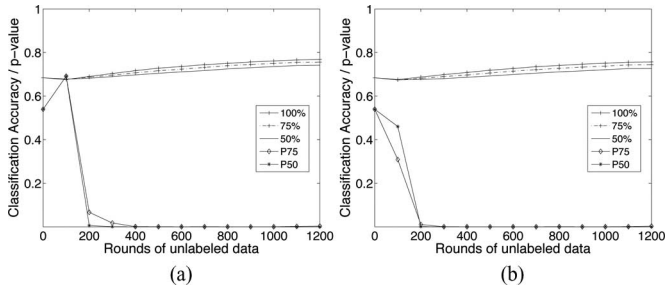

Fig. 1. CA versus rounds of unlabeled data for different percentages of classifier predictions used in self-/cotraining, for $l = 60$ and $n^R = 2$. P75 and P50 denotes the *P*-values for similar performance of 75% and 50% of most confident classifier predictions as compared to using 100%. (a) Cotraining. (b) Self-training.

The validation scheme is applied for all (l, n^R) combinations, maintaining the same orders of permutations of the data in each case.

IV. Results and Discussion

The algorithms were run for eight different configurations of the initial training data (l) and the number of rounds (n^R) used for the detection of each character. They are (l, n^R) - (40, 1), (60, 1), (60, 2), and (300, 2). The measure that is used to determine the most confident instances for retraining the classifier is the z-score, given in (14). Analysis of self-training and cotraining using 50%, 75%, and 100% of most confident predictions for updating the classifier was done. A sample result when $l = 60$ and $n^R = 2$, averaged over all the subjects is given in Fig. 1. The results clearly demonstrate a better performance when all the labels are used for retraining the classifiers. The statistical significance of this conclusion is established from the low *p* values for the null hypothesis that using all the classifier predictions for self-/cotraining are not beneficial. A similar trend was observed for other configurations of (l, n^R) as well, for both self- and cotraining. Hence, all the results presented henceforth uses 100% of the classifier predictions for self-/co-training.

The cross-validated results for the four algorithms (SBLDA, SLDA, CBLDA, CLDA) are summarized in Figs. 2–8. In most of the discussions that follows, only SBLDA and CBLDA are included, as these almost always gave better results than their FLDA-based counterparts, the SLDA, and CLDA. Also, since our study is meant to highlight semisupervised learning in general, and co-training in particular, such a comparison would be more appropriate.

A. Effect of Training Data

It can be seen from the results that the increase in proportion of unlabeled data leads to a significant increase in the CA, especially in situations where relatively low amount of labeled data is available. This could be expected, as empirical studies have shown that the CA increases exponentially with labeled data and linearly with unlabeled data [34]. These results can be observed in Fig. 2. For all the five subjects, it can be observed that when the labeled data are sufficient, addition of unlabeled data

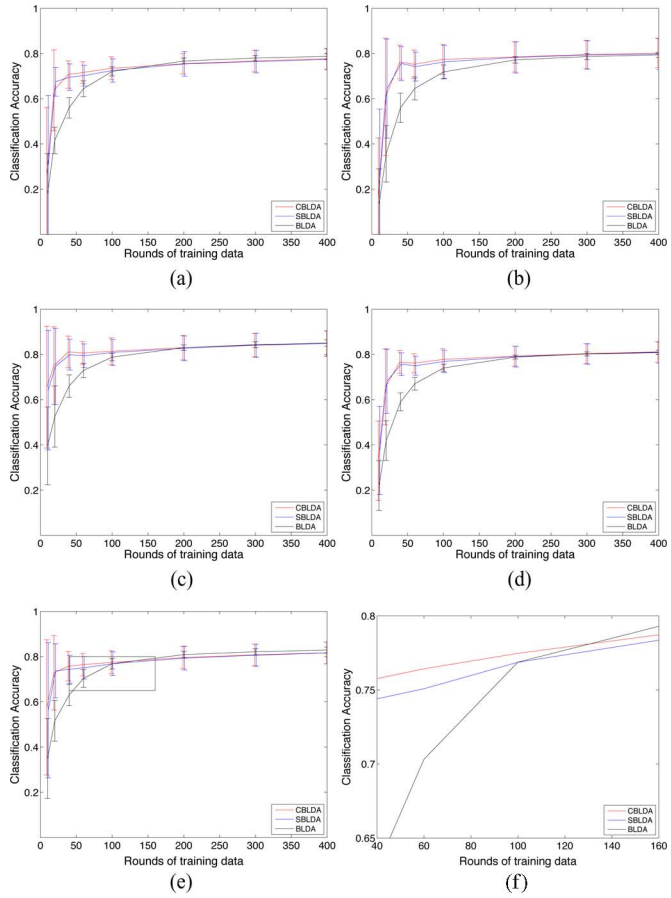

Fig. 2. CA of CBLDA, SBLDA, and fully supervised BLDA for various l (for $n^R = 2$), along with the error bars for $\pm\sigma$. (a) Subject 1. (b) Subject 2. (c) Subject 3. (d) Subject 4. (e) Subject 5. (f) Magnified version of data in the rectangle in Fig. 2(e).

does not improve the classifier performance. For all the subjects, approximately 200 rounds of data were enough to learn a classifier, which could not be improved further by semisupervised learning, and the CA using CBLDA bettered that of SBLDA in most cases.

B. Effect of Unlabeled Data

The effect of unlabeled data can be seen from Figs. 3–7 for various configurations of l and n^R. It can be seen that in most cases, the addition of unlabeled data helps increase the accuracy. However, as the ratio of unlabeled data to labeled data increases, the performance improvement decreases and gradually becomes minimal. A mathematical reasoning for this effect can be found in [19]. It can be seen in Figs. 3(e), 4(d), 5(d), 6(d), and 7(d) that the addition of unlabeled data when sufficient training data are available does not improve the classification performance of the system. For subjects 1 and 5, when $l = 300$ [see Fig. 3(e)], addition of unlabeled data in fact degrades the performance. This effect has been reported on semisupervised learning on different datasets by Cohen *et al.* [35].

In cases where there is an improvement, CBLDA almost always gives a better improvement over SBLDA. If unlabeled data were detrimental to classification performance (fourth group in

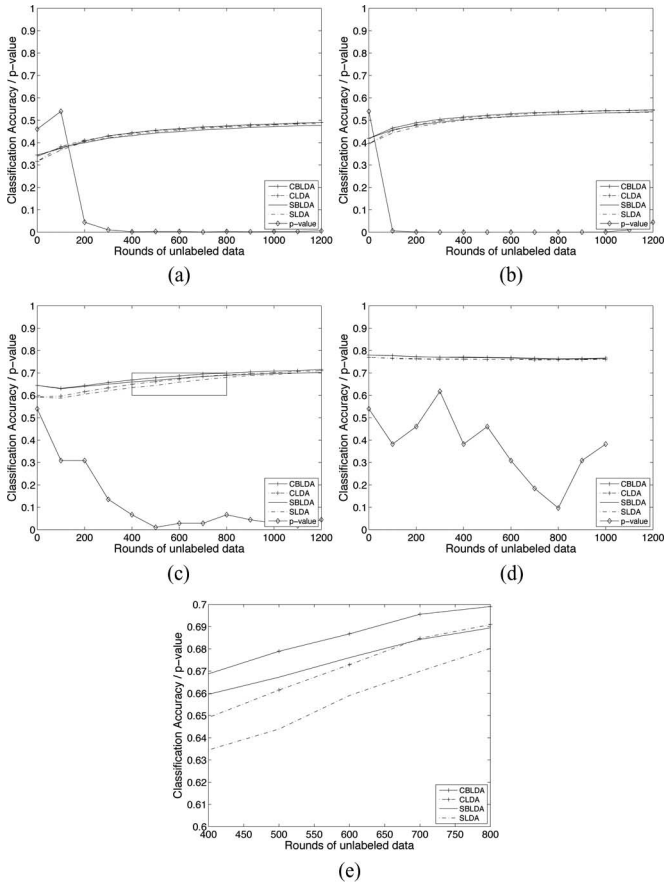

Fig. 3. CA versus rounds of unlabeled data for subject 1 for various l and n^R. (a) $l = 40$, $n^R = 1$. (b) $l = 60$, $n^R = 1$. (c) $l = 60$, $n^R = 2$. (d) $l = 300$, $n^R = 2$. (e) Magnified version of data in the rectangle in Fig. 3(c) (mean only).

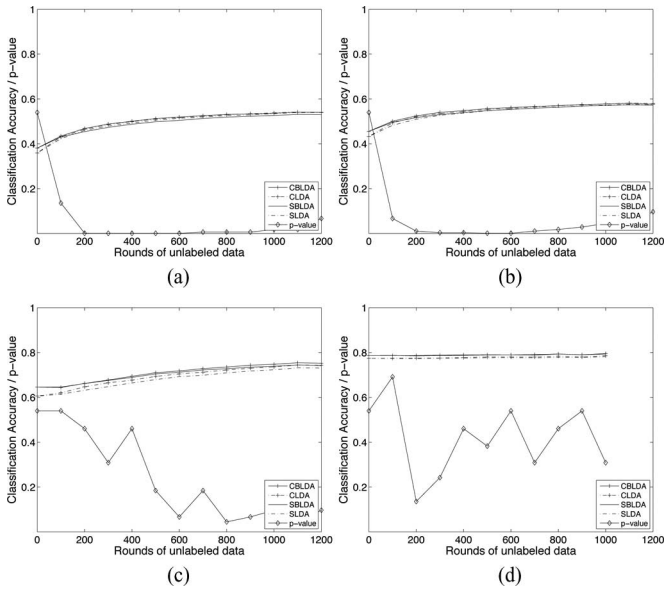

Fig. 4. CA versus rounds of unlabeled data for subject 2 for various l and n^R. (a) $l = 40$, $n^R = 1$. (b) $l = 60$, $n^R = 1$. (c) $l = 60$, $n^R = 2$. (d) $l = 300$, $n^R = 2$.

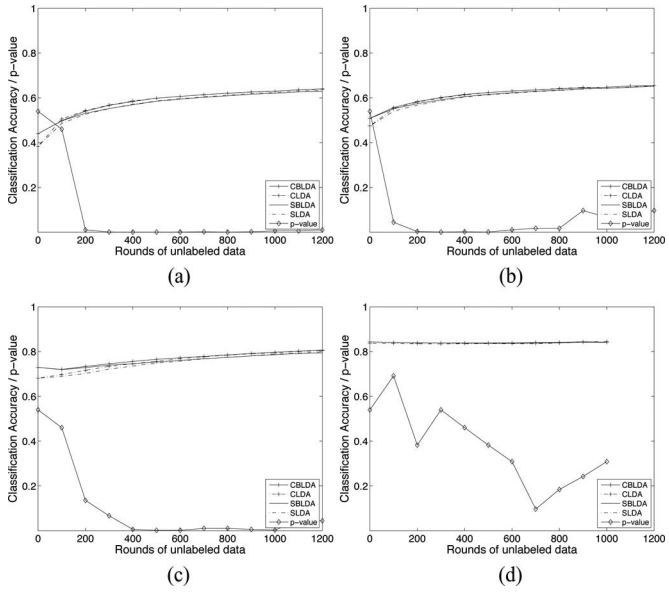

Fig. 5. CA versus rounds of unlabeled data for subject 3 for various l and n^R. (a) $l = 40$, $n^R = 1$. (b) $l = 60$, $n^R = 1$. (c) $l = 60$, $n^R = 2$. (d) $l = 300$, $n^R = 2$.

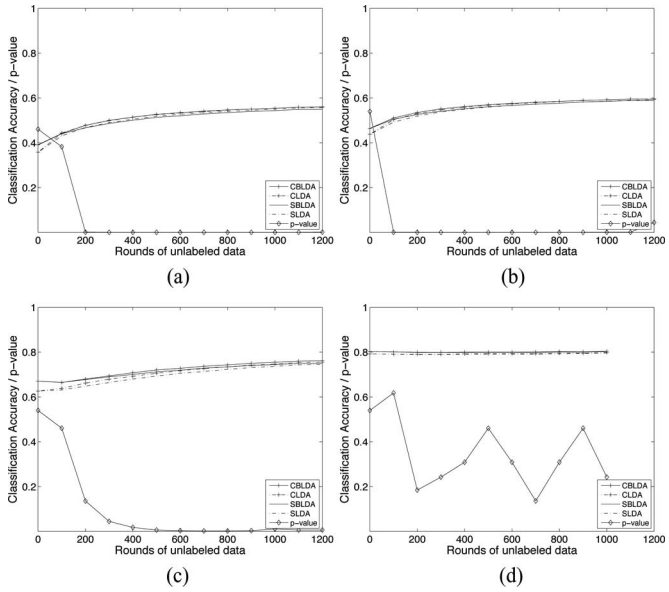

Fig. 6. CA versus rounds of unlabeled data for subject 4 for various l and n^R. (a) $l = 40$, $n^R = 1$. (b) $l = 60$, $n^R = 1$. (c) $l = 60$, $n^R = 2$. (d) $l = 300$, $n^R = 2$.

Fig. 8 for all subjects except subject 2), CBLDA reduced the initial accuracy only slightly as compared to SBLDA. Such a degradation can be observed in subjects 1 and 5 with the addition of unlabeled data when $l = 300$. A similar pattern is observed in the ITRs as well (see Fig. 8). It can be seen from Figs. 3–7 that the performance of CBLDA is significantly ($P < 0.05$) better than SBLDA for all subjects except when $l = 300$, in which case semisupervised learning offered no significant improvement over fully supervised classification. For the configuration ($l = 40, n^R = 1$), CBLDA gives a performance improvement of 13.2 bits/min more than supervised classifiers, and 1.7 bits/min

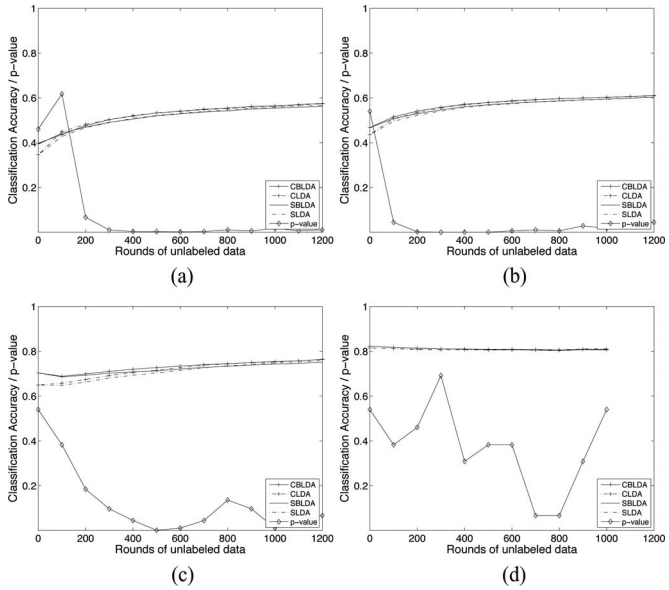

Fig. 7. CA versus rounds of unlabeled data for subject 5 for various l and n^R. (a) $l = 40$, $n^R = 1$. (b) $l = 60$, $n^R = 1$. (c) $l = 60$, $n^R = 2$. (d) $l = 300$, $n^R = 2$.

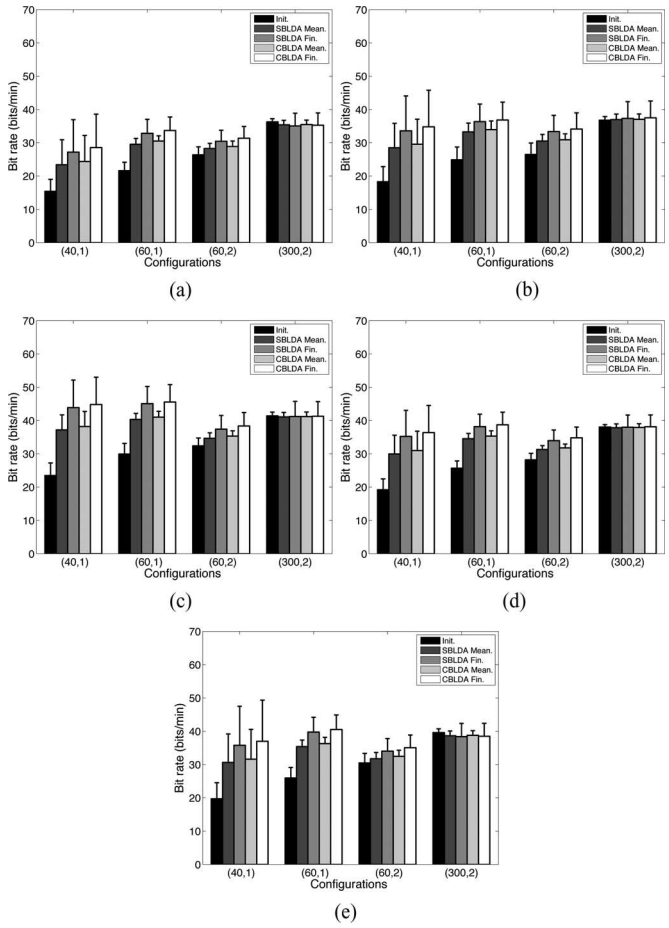

Fig. 8. Bar chart showing the bit rates for various configurations of l and n^R. The initial CA (Init.); as well as the mean classification (Mean.), and final classification accuracies (Fin.) achieved are shown for each (l, n^R) configuration and for each subject. (a) Subject 1. (b) Subject 2. (c) Subject 3. (d) Subject 4. (e) Subject 5.

more than SBLDA for subject 1; and 16.4 bits/min more than supervised classifiers, and 1.2 bits/min more than SBLDA for subject 2. The improvement is 21.0 and 17.2 bits/min over supervised classifiers, and 1.5 bits/min and 1.4 bits/min over SBLDA for subject 3 and 4, respectively. For subject 5, the algorithm achieved an increase of 18.5 bits/min over supervised classifiers and 1.6 bits/min over SBLDA. From these results, we can see that CBLDA outperforms SBLDA in most situations, though the actual amount of increase is not large. The final bit rates averaged over all the subjects is approximately 37 bits/min, which is 17 bits/min more than the initial CA. This was achieved with just 40 rounds of labeled data, which corresponds to a training time of about 90 s. This compares favorably with most state of the art BCI systems, where average bit rates of 30–40 bits/min are achieved with several minutes of training data [10], [36].

C. Subjectivity

The algorithm is found to have effective performance enhancement in all the five subjects. We can observe from the results that the algorithm is especially effective for subject 3, even when l is 40 and n^R is 1, which corresponds to a training time of less than 2 min. This could be due to the fact that subject 3 produces stronger P300 (which can be inferred from the fact that subject 3 gives the best performance with supervised classifiers, as can be seen from Fig. 2), and the prediction by both the classifiers are reasonably good. Consequently, the possibility of errors reinforcing themselves catastrophically is minimized, especially in cases where the training data are low.

For subject 2, the performance of CBLDA was not significantly better than that of SBLDA [see Fig. 4(c)] when $l = 60$, $n^R = 2$. For subject 1, when the training data are sufficient, degradation of accuracy with addition of unlabeled data are even more prominent [see Fig. 3(e), as compared to subjects 2, 3, and 4 [see Figs. 4(d), 5(d), and 6(d)]. Subject 2 always had an increase in performance even when training data were abundant [see Fig. 4(d)]. For other configurations, the enhancement of accuracy in subjects 2, 3, and 4 are found to be consistently better than subject 1. These are reflected in the ITRs as well. From these results, it can be concluded that the semisupervised learning is even more effective for subjects with a stronger P300, especially for the extreme cases of abundance and scarcity of training data.

D. Computational Complexity

Both self-training and cotraining are generalized methods, and the exact complexity depends on the particular classification algorithms used in their realization. For self-training, the complexity is related to the complexity of the specific classifier used, whereas for cotraining, it is determined by the sum of the complexities of the two classifiers used. In both cases, as more and more unlabeled data are added, complexity increases, as the classifier has to be retrained from a bigger pool of data. In the cotraining described in this paper, CBLDA and SBLDA have comparable complexities due to the fact that BLDA complexity, while being data and stopping criterion dependent, is much higher than that of the FLDA on average. One complete

run (corresponding to iterative classification of approximately 1 h of preprocessed data) of the MATLAB code running on a Windows Vista desktop computer with a 2.8 -GHz dual-core processor and 4 GB of RAM takes approximately 13.2 and 12.7 s, respectively.

V. Conclusion

A two-classifier cotraining-based approach is proposed to train robust classifiers, using both labeled and unlabeled data. The difference between the two classifiers is exploited for delivering a performance, which is superior to that of single classifier systems. The algorithm is able to utilize unlabeled data effectively to improve the performance of the classifier. This leads to a reduction in the user effort, and consequently, results in a more convenient BCI system. Also, the proposed method is shown to outperform the self-training-based approaches in most situations.

The addition of unlabeled data was found to increase the CA to a limit, beyond which the improvement was minimal. Also, if sufficient training data are available, the performance improvement due to the algorithm is minimal, or even negative. Introducing artificial training examples [37] to preserve diversity might reduce the tendency of cotraining algorithms to degenerate to self-training with the addition of more and more unlabeled data. Also, the use of more classifiers and a majority voting procedure could be used to determine the winner, thereby leveraging the advantages of both ensemble learning as well as cotraining.

In practical situations, there may be gradual changes in the data with factors such as gel drying, changes in cognitive state of patient, and adding unlabeled data might help in gradual adaptation of the classifier. In the cross-validations used in this paper, such adaptation effects are ignored. For getting a clearer picture on the performance of the proposed scheme in practical situations, extensive experiments have to be done on a wider user base without having to rely on cross-validation schemes. Moreover, when the training data are low and based on a few characters, the classifier might have been trained on data which are subject to visual attention and spatial gradient effects [38], which can introduce a bias in the result. With a carefully chosen initial training pattern, the initial classifier prediction will be less subject to biases, and the performance is likely to improve.

References

[1] G. Dornhege, J. D. R. Millan, T. Hinterberger, D. J. Mcfarland, K.-R. Muller, and T. J. Sejnowski, *Toward Brain-Computer Interfacing*. Cambridge, MA: MIT Press, 2007.

[2] L. A. Farwell and E. Donchin, "Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials," *Electroenceph. Clin. Neurophysiol.*, vol. 70, pp. 510–523, 1988.

[3] N. Xu, X. Gao, B. Hong, X. Miao, S. Gao, and F. Yang, "BCI competition 2003-data set IIb: Enhancing P300 wave detection using ICA-based subspace projections for BCI applications," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1067–1072, Jun. 2004.

[4] B. Allison and J. Pineda, "ERPs evoked by different matrix sizes: Implications for a brain–computer interface (BCI) system," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 11, no. 2, pp. 110–113, Jun. 2003.

[5] F. Nijboer, E. W. Sellers, J. Mellinger, M. A. Jordan, T. Matuz, A. Furdea, S. Halder, U. Mochty, D. J. Krusienski, T. M. Vaughan, J. R. Wolpaw, N. Birbaumer, and A. Kübler, "A P300-based brain-computer interface for people with amyotrophic lateral sclerosis," *Clin. Neurophysiol.*, vol. 119, pp. 1909–1916, Aug. 2008.

[6] A. Kübler, B. Kotchoubey, J. Kaiser, J. R. Wolpaw, and N. Birbaumer, "Brain–computer communication: Unlocking the locked in," *Psychol. Bull.*, vol. 127, no. 3, pp. 358–375, May 2001.

[7] E. Donchin, K. Spencer, and R. Wijesinghe, "The mental prosthesis: Assessing the speed of a P300-based brain-Computer Interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 8, no. 2, pp. 174–179, Jun. 2000.

[8] F. Lotte, M. Congedo, A. Lecuyer, F. Lamarche, and B. Arnaldi1, "A review of classification algorithms for EEG-based Brain Computer Interfaces," *J. Neural Eng.*, vol. 4, pp. R1–R13, 2007.

[9] A. Bashashati, M. Fatourechi, R. K. Ward, and G. E. Birch, "A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals," *J. Neural Eng.*, vol. 4, pp. 32–57, 2007.

[10] A. Lenhardt, M. Kaper, and H. Ritter, "An adaptive P300-based online brain-computer interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 2, pp. 121–130, Apr. 2008.

[11] P. Sykacek, S. Roberts, and M. Stokes, "Adaptive BCI based on variational Bayesian Kalman filtering: an empirical evaluation," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 5, pp. 719–727, May 2004.

[12] M. Thulasidas, C. Guan, and J. Wu, "Robust classification of EEG signal for Brain-Computer Interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 1, pp. 24–29, Mar. 2006.

[13] J. Blumberg, J. Rickert, S. Waldert, A. Schulze-Bonhage, A. Aertsen, and C. Mehring, "Adaptive classification for brain-computer Interfaces," in *Proc. 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2007, pp. 2536–2539.

[14] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1995, pp. 231–238.

[15] X. Liao, D. Yao, and C. Li, "Transductive SVM for reducing the training effort in BCI," *J. Neural Eng.*, vol. 4, no. 3, pp. 246–254, 2007.

[16] Y. Li, H. Li, C. Guan, and Z. Chin, "A self-training semi-supervised support vector machine algorithm and its applications in brain computer interface," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP 2007)*, Apr. 15–20, vol. 1, pp. I-385–I-388.

[17] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annual Conf. Computational Learning Theory (COLT 1998)*. New York: ACM, pp. 92–100.

[18] S. Goldman and Y. Zhou, "Enhancing supervised learning with unlabeled data," in *Proc. 17th Int. Conf. Machine Learning (ICML 2000)* June 29–July 2, pp. 327–334.

[19] W. Wang and Z.-H. Zhou, "Analyzing co-training style algorithms," in *Proc. 18th Eur. Conf. Machine Learning (ECML 2007:)*. Berlin, Heidelberg, Germany: Springer-Verlag, pp. 454–465.

[20] Z.-H. Zhou and M. Li, "Semisupervised regression with cotraining-style algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 11, pp. 1479–1493, Nov. 2007.

[21] D. J. Krusienski, E. W. Sellers, F. Cabestaing, S. Bayoudh, McFarland, M. Vaughan, and J. R. Wolpaw, "A comparison of classification techniques for the P300 speller," *J. Neural Eng.*, vol. 3, pp. 299–305, 2006.

[22] A. Rakotomamonjy and V. Guigue, "BCI competition III: Dataset II-ensemble of SVMs for BCI P300 speller," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 3, pp. 1147–1154, Mar. 2008.

[23] U. Hoffmann, J. M. Vesin, T. Ebrahimi, and K. Diserens, "An efficient P300-based Brain-Computer Interface for disabled subjects," *J. Neurosci. Meth.*, vol. 167, no. 1, pp. 115–125, 2008.

[24] X. Lei, P. Yang, and D. Yao, "An empirical Bayesian framework for Brain-Computer Interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 17, no. 6, pp. 521–529, Dec. 2009.

[25] S. T. Ahi, H. Kambara, and Y. Koike, "A comparison of dimensionality reduction techniques for the P300 response," in *Proc. 3rd Int. Conv. Rehabil. Eng. Assistive Technol. (i-CREATe '09:)*. New York: ACM, pp. 1–4.

[26] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley, 2001.

[27] H. Zhang, C. Guan, and C. Wang, "Asynchronous P300-based brain-computer interfaces: A computational approach with statistical models," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 6, pp. 1754–1763, Jun. 2008.

[28] D. MacKay, "Bayesian interpolation," *Neural Comput.*, vol. 4, no. 3, pp. 415–417, 1992.

[29] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Lear. Res.*, vol. 1, pp. 211–244, 2001.

[30] J. Wolpaw, N. Birbaumer, W. Heetderks, D. McFarland, P. Peckham, G. Schalk, E. Donchin, L. Quatrano, C. Robinson, and T. Vaughan, "Brain-Computer Interface technology: A review of the first international

meeting," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 8, no. 2, pp. 164–173, Jun. 2000.
[31] W. Mendenhall, D. D. Wackerly, and R. L. Scheaffer, Eds., *Mathematical Statistics With Applications*. Boston, MA: PWS-Kent, 1989.
[32] Advanced Neuro Technology Website. [Accessed: May 2, 2010]. [Online]. Available: http://www.ant-neuro.com/
[33] G. H. Klem, H. O. Lüders, H. H. Jasper, and C. Elger, "The ten-twenty electrode system of the International Federation. The International Federation of Clinical Neurophysiology," *Electroenceph. Clin. Neurophysiol.*, vol. 52, pp. 3–6, 1999.
[34] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006.
[35] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang, "Semisupervised learning of classifiers: Theory, algorithms, and their application to Human-Computer Interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 12, pp. 1553–1567, Dec. 2004.
[36] E. Donchin and Y. Arbel, "P300 based brain computer interfaces: A progress report," presented at the 5th Int. Conf. Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience. Berlin, Heidelberg, Germany: Springer-Verlag, 2009, pp. 724–731.
[37] P. Melville and R. J. Mooney, "Constructing diverse classifier ensembles using artificial training examples," presented at the 18th Int. Joint Conf. Artificial intelligence (IJCAI 2003). San Francisco, CA: Morgan Kaufmann, pp. 505–510.
[38] I. Shevelev, E. Mikhailova, V. Chicherov, V. Konishev, and D. Karlovskiy, "Spatial gradient of P300 in the brain-computer interface paradigm," *Int. J. Psychophysiol.*, vol. 69, no. 3, pp. 181–181, 2008.

Rajesh C. Panicker (S'09) received the B.Tech degree in electronics and communication engineering from the University of Kerala, Thiruvananthapuram, India, in 2005. He is currently working toward the Ph.D. degree at the Department of Electrical and Computer Engineering, National University of Singapore.

His current research interests include brain-computer interfaces, signal processing, and pattern recognition.

Sadasivan Puthusserypady (M'00–SM'05) received the B.Tech degree in electrical engineering, in 1986, and the M.Tech degree in instrumentation and control systems engineering, in 1989, both from the University of Calicut, India, and the Ph.D. degree in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1995.

During 1993 through 1996, he was a Research Associate at the Department of Psychopharmacology, National Institute of Mental Health and Neuro Sciences, Bangalore, India. From 1996 to 1998, he was a Postdoctoral Research Fellow at the Communications Research Laboratory, McMaster University in Hamilton, ON, Canada. In 1998, he joined Raytheon Systems Canada Ltd., Waterloo, ON, as a Senior Systems Engineer. In 2000, he moved to the National University of Singapore, as an Assistant Professor in the Department of Electrical and Computer Engineering until 2009. He is currently an Associate Professor at the Department of Electrical Engineering, Technical University of Denmark, Lyngby, Denmark. His current research interests include biomedical signal processing, brain–computer interfaces, and home health care systems.

Ying Sun (S'00–M'05) received the B.Eng. degree from Tsinghua University, Beijing, China, in 1998, the M.Phil. degree from Hong Kong University of Science and Technology, Hong Kong, in 2000, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2004.

She was a Member of Technical Staff in the Imaging and Visualization Department, Siemens Corporate Research, Princeton, NJ. Since 2007, she has been with the Department of Electrical and Computer Engineering, National University of Singapore, where she is currently an Assistant Professor. Her research interests include medical image analysis, signal processing, and pattern recognition.